

## Processing a text file in a computer application

```
forming a template from
fragments of the text file
```

```
using the template as an
overlay for parsing incoming
files, or as a prototype to
generate a segment of an
output file
```

using a macro class to map  
data from the text file to an  
application

```
embedding the macro class as a
keyword within the template,
wherein when the template
reaches the keyword, the
template calls the macro class
to further process the text
file
```

Fig. 1

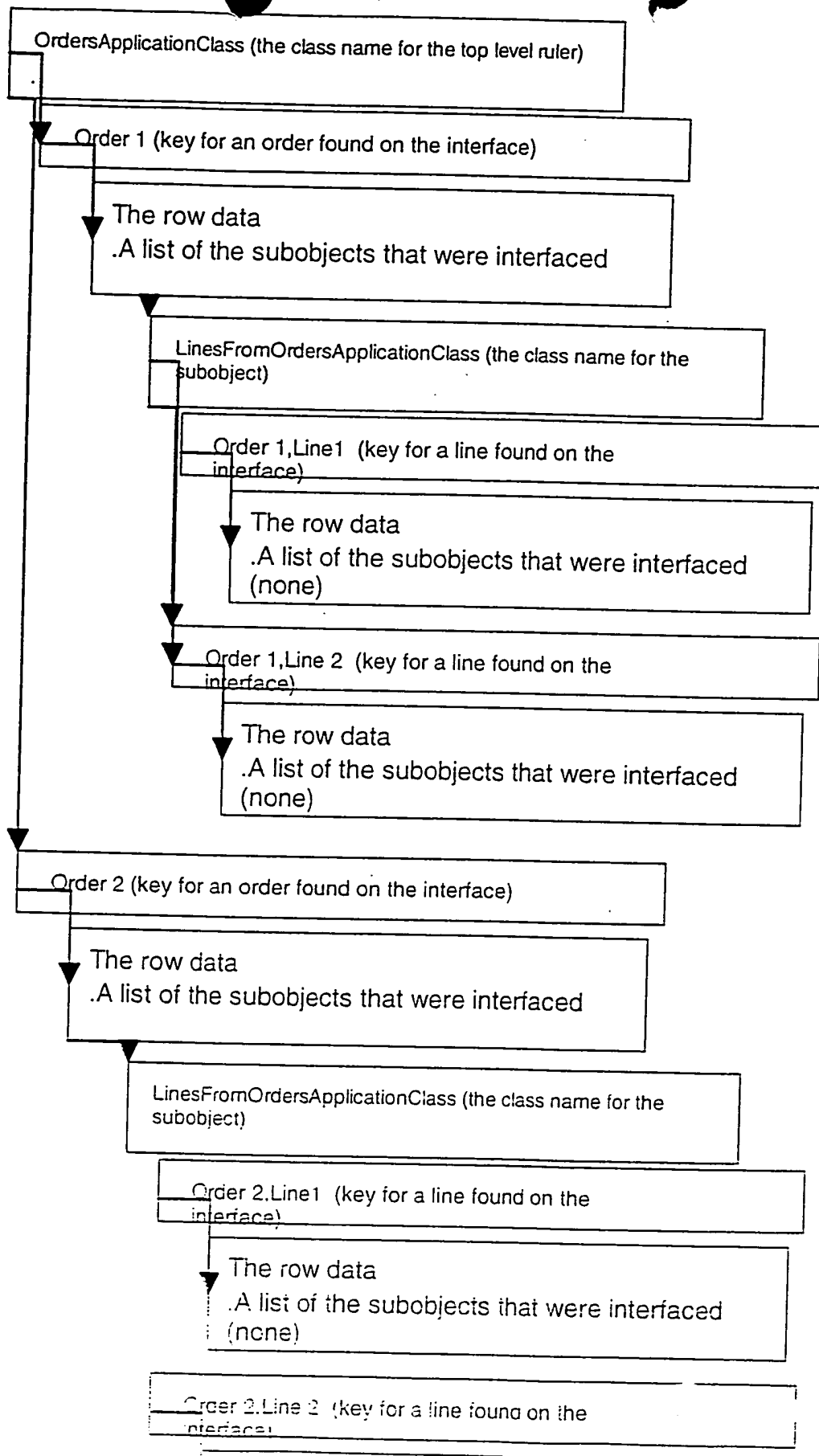


Fig. 3

1. Templates that support variable substitution and conditional or iterative generation for output files
2. A mechanism to fill in variable substitutions in the templates
3. A mechanism to handle iterations and conditional processing
4. A flexible template processing mechanism which operates within an object oriented environment
5. A flexible and easily extended formatting mechanism for output
6. A mechanism for template creation and editing
7. Special support for complex object presentation
8. Templates that support parsing of the input file and mapping of its contents into a complex object structure suitable for update processing
9. A mechanism to extract data from the input stream
10. A mechanism to map the structure of the input file into a complex object structure suitable for internal processing
11. A mechanism to organize the updating data from the input file and to apply it within the application
12. A flexible and easily extended formatting mechanism
13. A mechanism for input template creation and editing.

Fig. 2

00000000-00000000